**МИНИСТЕРСТВО ТРАНСПОРТА**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ**

**АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО**

**ОБРАЗОВАНИЯ**

**“РОССИЙСКИЙ УНИВЕРСИТЕТ**

**ТРАНСПОРТА” (МИИТ)**

**Институт транспортной техники и систем**

**управления**

**(ИТТСУ)**

**Кафедра “Управление и защита информации”**

**Тема: “Лабораторная работа №3”**

**Выполнили студенты группы ТКИ-342**

**Павлючков Даниил Дмитриевич**

**Рудов Лев Андреевич**

**Специальность 10.05.01 “Компьютерная безопасность”**

**Преподаватель: к.т.н. Сафронов А.И.**

**Москва**

**2024**

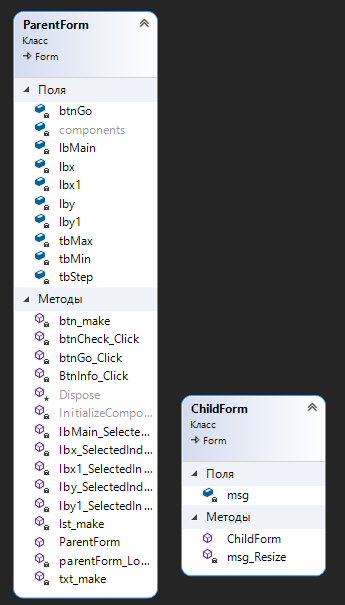
1. Цель работы:

Закрепить навыки разработки визуального пользовательского интерфейса, освоить работу с текстовыми файлами в среде *Microsoft Visual Studio*, научиться взаимно увязывать одни элементы управления с другими, получить представление о перерисовке и перемасштабировании

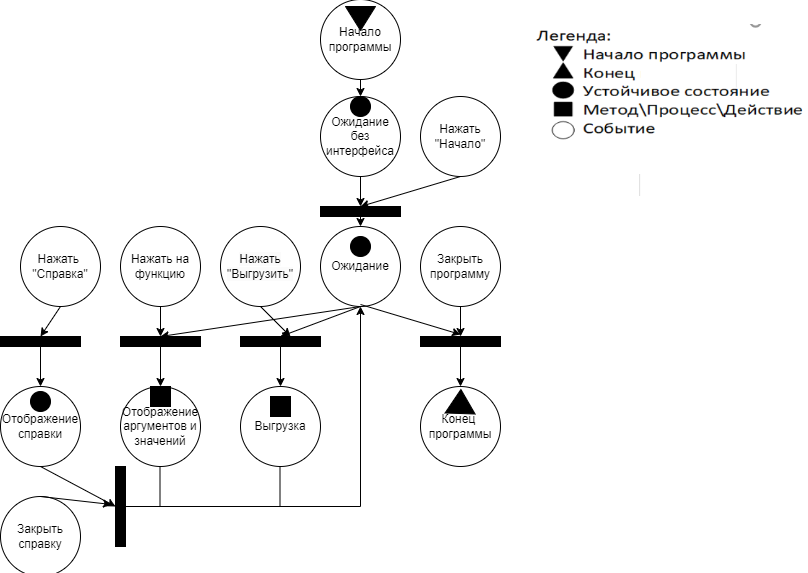
2. Задача:

В интегрированной среде разработки *Microsoft Visual Studio* на языке *Visual C#* разработать графический пользовательский интерфейс, состоящий из двух экранных форм (основной и дочерней), все элементы управления которого полностью настраиваются кодом без использования графического конструктора».

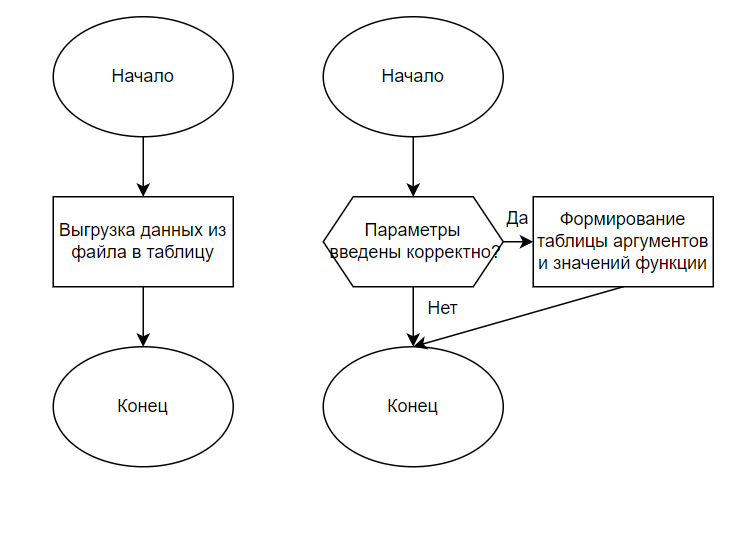
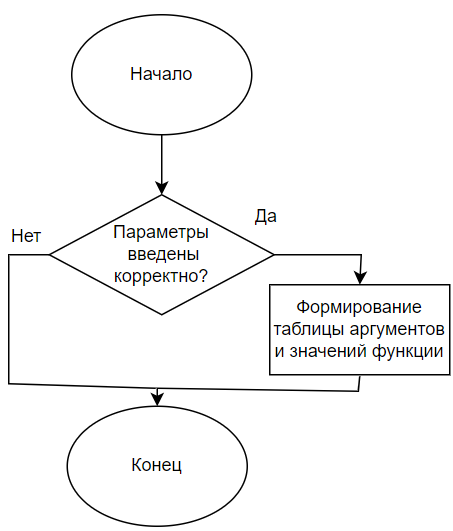
3. Диаграмма классов:



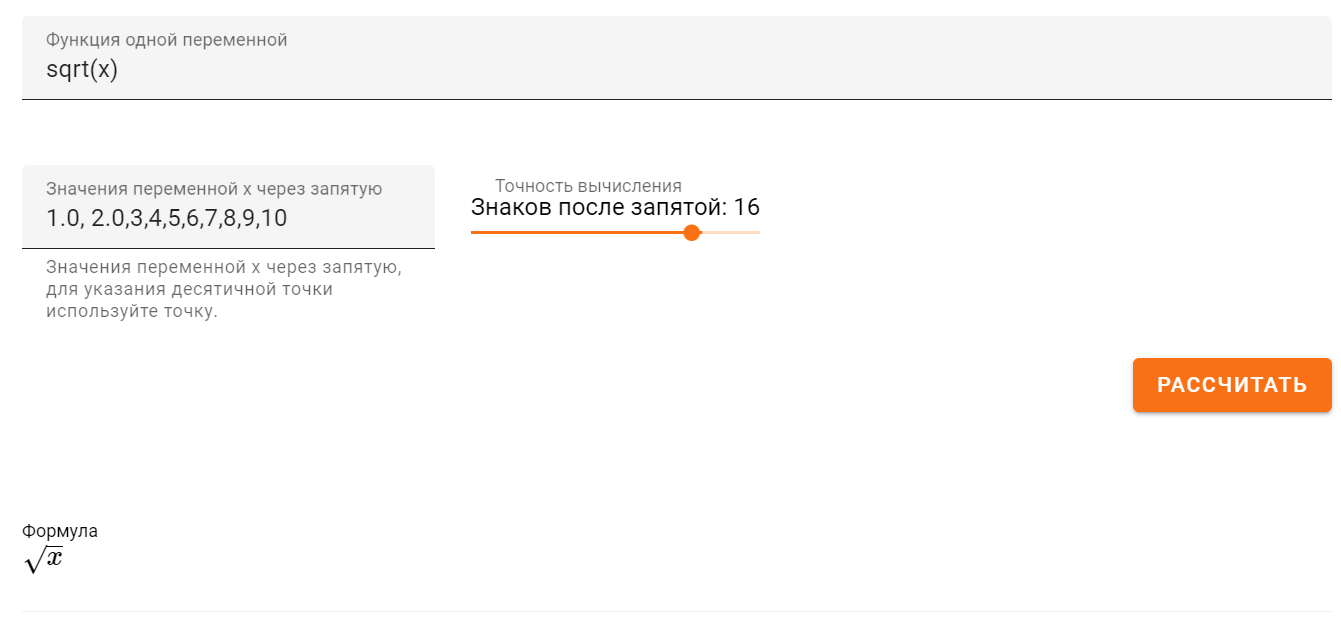
4. Сеть Петри

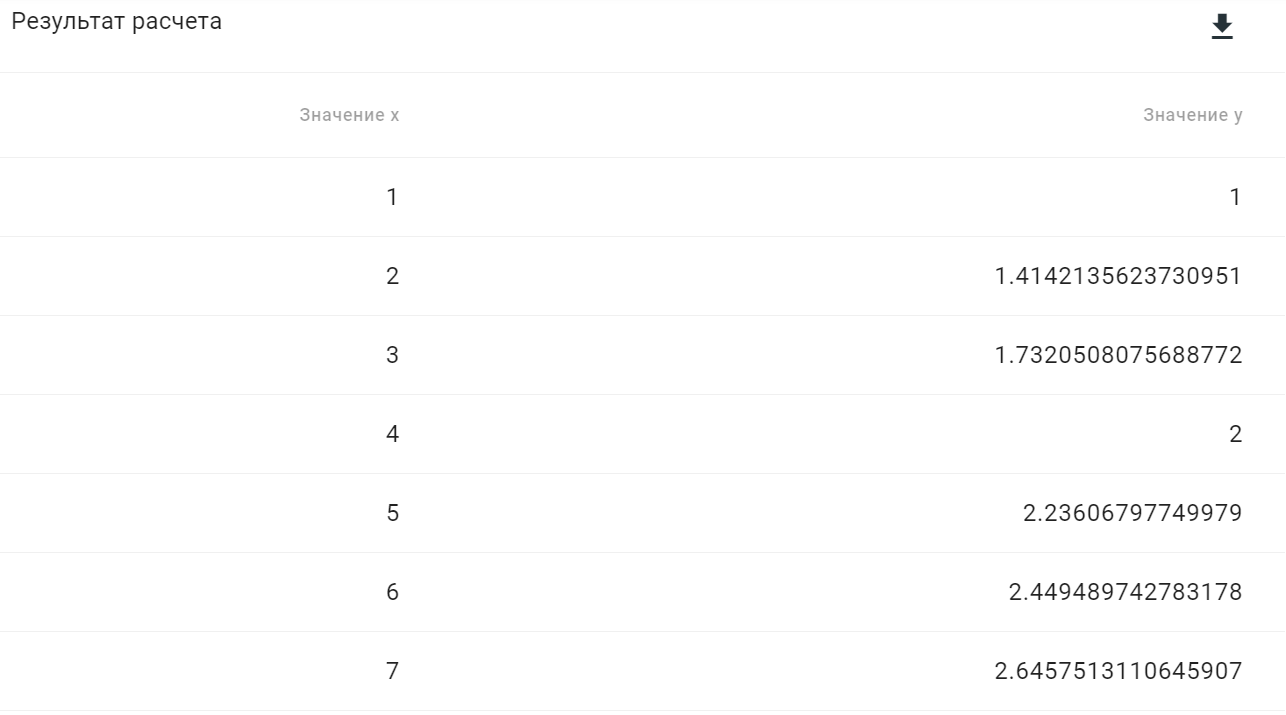


5. Схемы алгоритмов методов:



6. Подбор тестовых примеров





7. Листинг кода:

using System;

using System.Drawing;

using System.IO;

using System.Windows.Forms;

namespace Lab3

{

public partial class ParentForm : Form

{

public ParentForm()

{

InitializeComponent();

}

private Button btnGo;

private TextBox tbMin;

private TextBox tbMax;

private TextBox tbStep;

private ListBox lby;

private ListBox lbx;

private ListBox lbx1;

private ListBox lby1;

private ListBox lbMain;

private void parentForm\_Load(object sender, EventArgs e)

{

btn\_make();

txt\_make();

lst\_make();

btnGo.Visible = true;

}

private void btn\_make()

{

BackColor = SystemColors.Info;

Cursor = Cursors.Hand;

Button Buttongenerator(int x, int y, string label, bool anchor)

{

Button example = new Button();

example.Location = new Point(this.Width - x, this.Height - y);

example.Text = label;

example.AutoSize = true;

example.BackColor = Color.Black;

example.Padding = new Padding(6);

example.Font = new Font("Times New Roman", 18);

example.Visible = false;

example.ForeColor = Color.White;

if (anchor == true) example.Anchor = (AnchorStyles.Top | AnchorStyles.Right);

return example;

}

Button btnInfo = Buttongenerator(100, 485, "Справка", true);

btnInfo.Click += BtnInfo\_Click;

Controls.Add(btnInfo);

btnGo = Buttongenerator(800, 100, "Нажмите, чтобы начать работу", false);

btnGo.Click += btnGo\_Click;

Controls.Add(btnGo);

Button btnCheck = Buttongenerator(100, 100, "Выгрузить", true);

btnCheck.Click += btnCheck\_Click;

Controls.Add(btnCheck);

}

private void txt\_make()

{

TextBox TextBoxgenerator(int x, int y)

{

TextBox example = new TextBox();

example.Location = new Point(x, y);

example.AutoSize = true;

example.Visible = false;

return example;

}

tbMin = TextBoxgenerator(btnGo.Left, this.Height - 470);

tbMin.Text = "Нижняя граница";

Controls.Add(tbMin);

tbStep = TextBoxgenerator(btnGo.Left + btnGo.Width / 3, this.Height - 470);

tbStep.Text = "Шаг";

Controls.Add(tbStep);

tbMax = TextBoxgenerator(btnGo.Left + 2 \* btnGo.Width / 3, this.Height - 470);

tbMax.Text = "Вехняя граница";

Controls.Add(tbMax);

}

private void lst\_make()

{

ListBox ListBoxgenerator(int x, int y)

{

ListBox example = new ListBox();

example.Location = new Point(x, y);

example.Size = new Size(btnGo.Width / 3, 200);

example.Visible = false;

return example;

}

lbx = ListBoxgenerator(btnGo.Left + btnGo.Width / 3, 65);

lbx.SelectedIndexChanged += lbx\_SelectedIndexChanged;

Controls.Add(lbx);

lby = ListBoxgenerator(lbx.Right, 65);

lby.SelectedIndexChanged += lby\_SelectedIndexChanged;

Controls.Add(lby);

lbx1 = ListBoxgenerator(lby.Right, 65);

lbx1.Text = "Аргументы";

lbx1.SelectedIndexChanged += lbx1\_SelectedIndexChanged;

Controls.Add(lbx1);

lby1 = ListBoxgenerator(lbx1.Right, 65);

lby1.Text = "Функции";

lby1.SelectedIndexChanged += lby1\_SelectedIndexChanged;

Controls.Add(lby1);

lbMain = ListBoxgenerator(btnGo.Left, 65);

lbMain.Items.Add("Sin (x)");

lbMain.Items.Add("Log\_2 (x)");

lbMain.Items.Add("Cbrt(x^2) / (x^2 - 1)");

lbMain.Items.Add("ArcCos (x)");

lbMain.Items.Add("Sqrt (x)");

lbMain.Items.Add("Cos (x)");

lbMain.Items.Add("ArcSin (x)");

lbMain.Items.Add("Log\_e (x)");

lbMain.Items.Add("Log\_10 (x)");

lbMain.Items.Add("ArcTg (x)");

lbMain.Items.Add("x^2");

lbMain.Items.Add("Tan (x)");

lbMain.SelectedIndexChanged += lbMain\_SelectedIndexChanged;

Controls.Add(lbMain);

}

private void lbMain\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lbMain.SelectedIndex > -1)

{

lbx.Items.Clear();

lby.Items.Clear();

if (string.IsNullOrEmpty(tbMin.Text) || string.IsNullOrEmpty(tbMax.Text))

{

return;

}

bool success = double.TryParse(tbMin.Text, out double trash);

bool success1 = double.TryParse(tbMax.Text, out double trash1);

if (success == false || success1 == false)

{

return;

}

int counter = 0;

using (StreamWriter sw = new StreamWriter("Data.txt"))

{

double arg = Double.Parse(tbMin.Text);

double setstep = Double.Parse(tbStep.Text);

int top = Convert.ToInt32(Math.Ceiling((Double.Parse(tbMax.Text) - Double.Parse(tbMin.Text))));

int convstep = Convert.ToInt32(Math.Ceiling(Double.Parse(tbStep.Text)));

for (double i = arg; i <= top + 1; i = i+convstep)

{

lbx.Items.Add(arg);

switch (lbMain.SelectedIndex)

{

case 0:

lby.Items.Add(Math.Sin(i));

break;

case 1:

lby.Items.Add(Math.Log(i, 2));

break;

case 2:

if (i > 0.9999999 && i < 1.0000001) lby.Items.Add("Деление на 0");

else

lby.Items.Add((Math.Pow(i, (2.0 / 3.0))) / (i \* i - 1));

break;

case 3:

lby.Items.Add(Math.Acos(i));

break;

case 4:

lby.Items.Add(Math.Sqrt(i));

break;

case 5:

lby.Items.Add(Math.Cos(i));

break;

case 6:

lby.Items.Add(Math.Asin(i));

break;

case 7:

lby.Items.Add(Math.Log(i));

break;

case 8:

lby.Items.Add(Math.Log10(i));

break;

case 9:

lby.Items.Add(Math.Atan(i));

break;

case 10:

lby.Items.Add(i \* i);

break;

case 11:

lby.Items.Add(Math.Tan(i));

break;

}

arg = arg + setstep;

string x = lbx.Items[counter].ToString();

string y = lby.Items[counter].ToString();

sw.Write(x);

sw.Write("=");

sw.Write(y);

sw.Write("\n");

counter++;

}

sw.Close();

}

}

}

private void lby\_SelectedIndexChanged(object sender, EventArgs e)

{

lbx.SelectedIndex = lby.SelectedIndex;

}

private void lby1\_SelectedIndexChanged(object sender, EventArgs e)

{

lbx1.SelectedIndex = lby1.SelectedIndex;

}

private void lbx\_SelectedIndexChanged(object sender, EventArgs e)

{

lby.SelectedIndex = lbx.SelectedIndex;

}

private void lbx1\_SelectedIndexChanged(object sender, EventArgs e)

{

lby1.SelectedIndex = lbx1.SelectedIndex;

}

private void btnCheck\_Click(object sender, EventArgs e)

{

string round(string rounding)

{

double temp = Double.Parse(rounding);

temp = Math.Round(temp, 1);

rounding = temp.ToString();

return rounding;

}

lbx1.Items.Clear();

lby1.Items.Clear();

lbx1.Items.Add("Аргументы");

lby1.Items.Add("Значения функции");

lbx1.Visible = true;

lby1.Visible = true;

string text = File.ReadAllText("Data.txt");

char[] check = {'$', '='};

string xy = "$";

foreach (var ch in text)

{

string temp = "";

temp += ch;

if (xy.Substring(xy.Length - 1, 1).Equals("="))

{

xy = xy.Trim(check);

lbx1.Items.Add(xy);

xy = "$";

}

if (temp.Equals("\n"))

{

xy = xy.Trim(check);

if (xy.IndexOf(",") > -1)

{

xy = round(xy);

}

lby1.Items.Add(xy);

xy = "$";

}

xy += ch;

}

}

private void btnGo\_Click(object sender, EventArgs e)

{

foreach (Control cntrl in Controls)

{

cntrl.Visible = true;

}

lbx1.Visible = false;

lby1.Visible = false;

}

private void BtnInfo\_Click(object sender, EventArgs e)

{

ChildForm Child = new ChildForm();

Child.Owner = this;

Child.Text = "Справка";

Child.ShowDialog();

Child.Dispose();

}

}

public partial class ChildForm : Form

{

private TextBox msg;

public ChildForm()

{

msg = new TextBox();

msg.Location = new Point(0, 0);

msg.Size = new Size(this.Width, this.Height);

msg.Multiline = true;

msg.Font = new Font("Times New Roman", 18);

msg.ReadOnly = true;

msg.Text = "Данная программа реализует автоматическое построение значений абсцисы и ординаты функции\n" +

"на промежутке между двумя задаваемыми значениями с шагом 1. Пожалуйста, введите значение\n" +

"нижней границы промежутка в левое заполняемое поле, а значение верхней границы промежутка\n" +

"в правое, после чего выберите одну из доступных функций и нажмите на кнопку (Нажмите, чтобы\n" +

"начать работу). В результате левый столбец будет заполнен значениями абсциссы, а правый\n" +

"значениями ординаты. Результаты автоматически выгружаются в файл Data.txt откуда их можно\n" +

"выгрузить обратно в программу с округлением по нажатию кнопки выгрузить. Результаты будут в отдельной таблице\n";

Controls.Add(msg);

Resize += msg\_Resize;

}

private void msg\_Resize(object sender, EventArgs e)

{

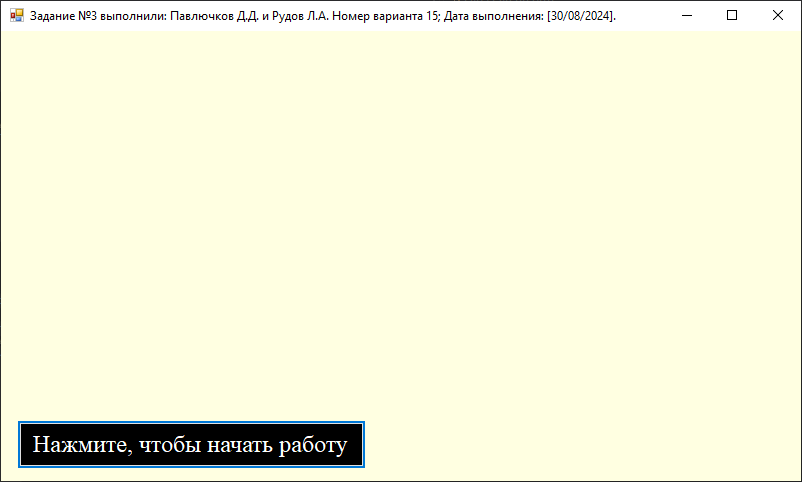
msg.Size = new Size(this.Width, this.Height);

}

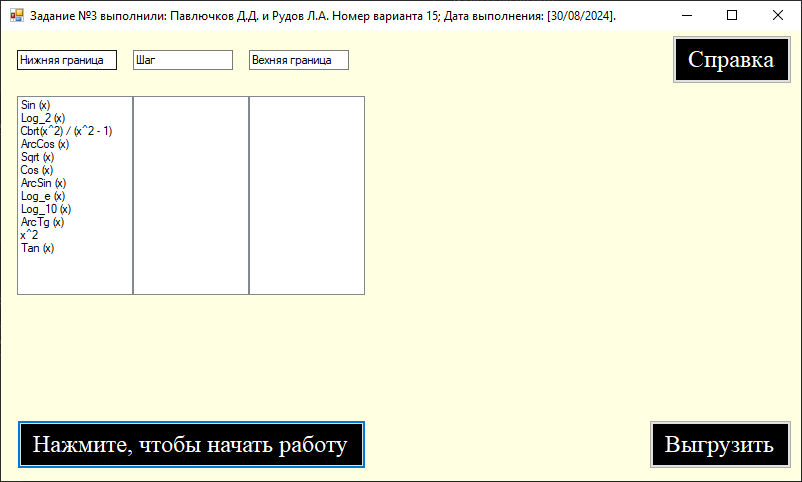
}

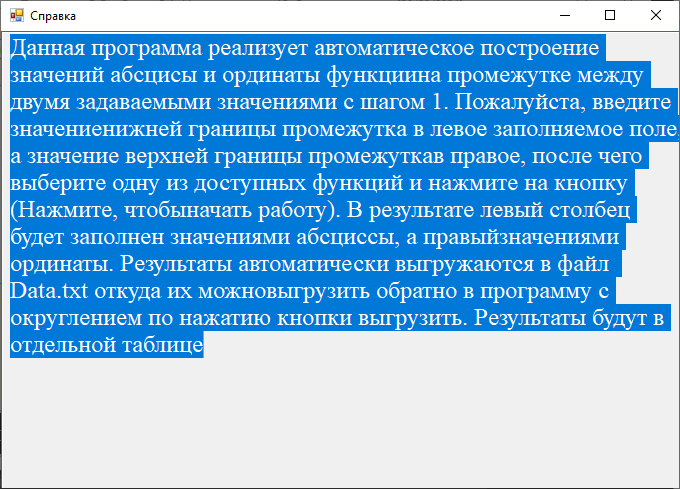
}

1. **Графический пользовательский интерфейс**

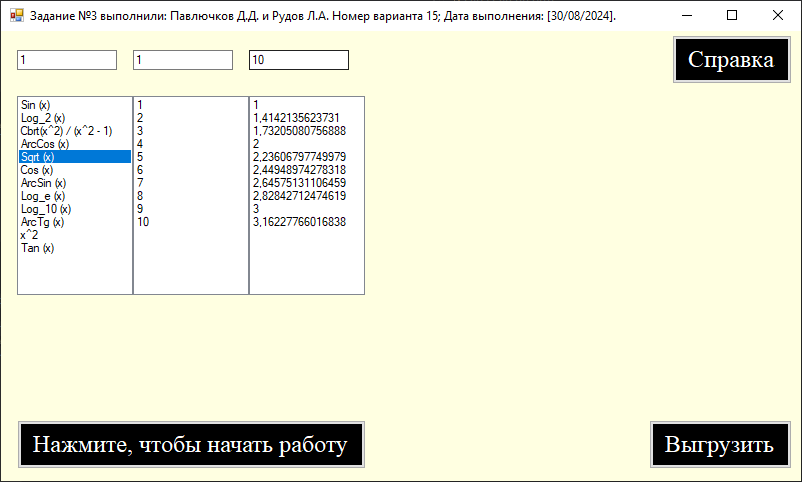
****

ГПИ стартовой страницы

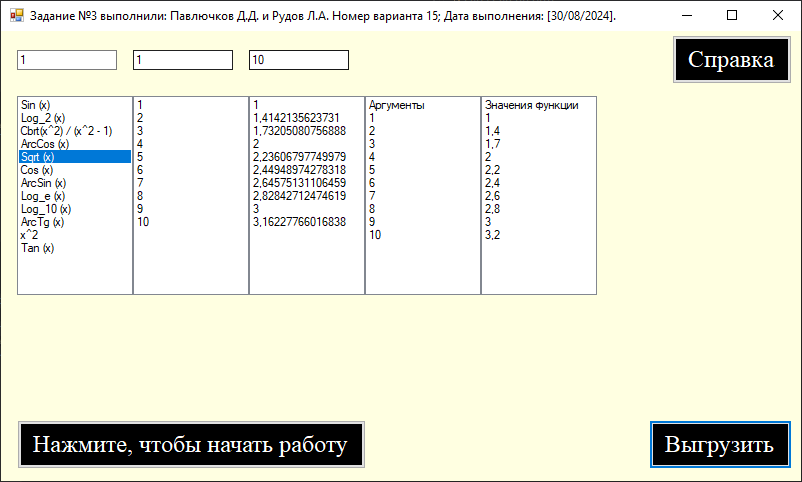
****

ГПИ основной страницы после начала работы с программой****

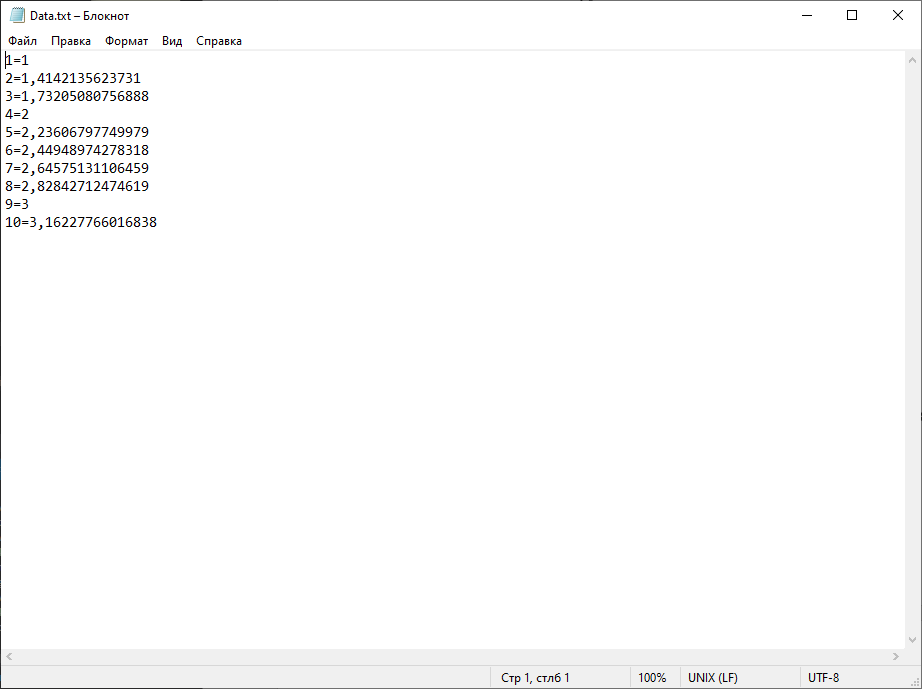
Справка, вызываемая нажатием на кнопку «Справка»



Результат работы программы с указанными функцией и границами



Результат работы программы с указанными функцией и границами после нажатия кнопки Выгрузить



Содержимое файла Data.txt, в который производится запись значений.

1. **Вывод**

В ходе выполнения лабораторной работы были закреплены навыки разработки визуального пользовательского интерфейса, освоена работа с текстовыми файлами в среде *Microsoft Visual Studio*, получены навыки работы с элементами управления.